Beyond Musical Descriptors: Extracting Preference-Bearing Intent in Music Queries

Marion Baranes[1], Romain Hennequin[1] and Elena V. Epure[1, 2]
research@deezer.com

[1]Deezer Research, Paris, France,
[2]Idiap Research Institute, Martigny, Switzerland.

Abstract

Although annotated music descriptor datasets for user queries are increasingly common, few consider the user's intent behind these descriptors, which is essential for effectively meeting their needs. We introduce MusicRecoIntent, a manually annotated corpus of 2,291 Reddit music requests, labeling musical descriptors across seven categories with positive, negative, or referential preference-bearing roles. We then investigate how reliably large language models (LLMs) can extract these music descriptors, finding that they do capture explicit descriptors but struggle with context-dependent ones. This work can further serve as a benchmark for fine-grained modeling of user intent and for gaining insights into improving LLM-based music understanding systems.

1 Introduction

Users increasingly expect machines to understand complex and subjective natural-language queries for music search or recommendation (Doh et al., 2023; Gupta et al., 2023; Porcaro and Saggion, 2019; Delcluze et al., 2025; Palumbo et al., 2025; Melchiorre et al., 2025). While search engines reliably handle focused queries—those naming a specific artist, track, or album—they remain much less effective for open queries where user intent is exploratory (Hosey et al., 2019; Sguerra et al., 2022). Addressing open queries requires interpreting complex musical descriptors like genre (e.g., *pop, jazz*), mood (e.g., *sad, calm*), listening context (e.g., *party, driving*), instrumentation (e.g., *guitar, piano*), time period (e.g., *80s*), or geographical origin (e.g., *Spanish music*).

However, understanding descriptors alone is insufficient: it is also necessary to determine how each relates to the user's intent—whether it expresses a desired attribute (e.g., *I want to listen some Rock*), an undesired one (e.g., *Recommend me anything except Elvis Presley.*), or serves as a reference point (e.g., similarity: *I want more recent music, but like Elvis Presley.*). We capture this distinction by assigning each descriptor a **preference-bearing intent**, expressing either *positive* affinity (+), *negative* aversion (-), or a *referential* role with softer notions of similarity (∼). Although several works (Hachmeier and Jäschke, 2025; Salganik et al., 2025a) focus on music descriptor extraction or annotation, none consider the associated preference-bearing intents, and it remains unclear whether LLMs can robustly capture both the breadth of musical attributes and the preference-bearing roles of these descriptors.

To fill these gaps, we propose: (1) *MusicRecoIntent*, an annotated corpus of music-related queries, with each descriptor linked to a preference-bearing role, enabling fine-grained analysis of user intents.[1] (2) A benchmark of popular LLMs for this extraction task. (3) A qualitative analysis of systematic errors in manual and automatic annotations.

2 Related Work

Several datasets focus on music metadata extraction, particularly named entities (NE) such as artists or track titles. Epure and Hennequin (2022) created *MusicRecoNER*, a Reddit-based corpus of music recommendation queries, showing that extracting musical entities is non-trivial. Hachmeier and Jäschke (2024, 2025) extended this work using LLMs on queries extracted from Reddit, but also YouTube. Other studies address semantic aspects of musical descriptions. Salganik et al. (2025a,b) introduced MusicSem, a large language–audio corpus annotated across five semantic categories, while Weck et al. (2024) extract descriptors such as genre, style, mood, instrumentation, and tempo from Wikipedia. Additional datasets, like MusicCaps (Agostinelli et al., 2023) or JamendoMax-

[1]Dataset is available at https://github.com/deezer/MusicRecoIntent-NLP4MusA26.

Proceedings of the 4th Workshop on NLP for Music and Audio (NLP4MusA 2026), pages 20–26

Caps (Roy et al., 2025), provide captions and aspect lists for musical segments, linking audio tracks with semantic descriptions.

Information Extraction (IE) aims to extract structured information from unstructured text, encompassing tasks such as Named Entity Recognition. In the musical domain, Hachmeier and Jäschke (2025) show that LLMs outperform smaller models in detecting musical entities, although performance strongly depends on prior exposure of LLMs to the entities. Similarly, Salganik et al. (2025b) demonstrate that LLMs can extract detailed semantic information from music descriptions, highlighting their growing adoption in music-related IE.

Beyond IE, determining the preference-bearing role of each descriptors remains challenging. Negation detection studies indicate that LLMs often struggle, where semantically contradictory statements are treated as equivalent (Kim et al., 2025; Vrabcová et al., 2025). Similarity detection in user musical queries has been shown by Palumbo et al. (2025) to be reliably handled by LLMs.

Despite recent advances, existing datasets and solutions fail to capture complex user intentions, including negation, or softer notions of similarity. This work fills these gaps by introducing a dataset for benchmarking LLMs' ability to model musical descriptors and their preference-bearing roles, supporting a richer understanding of user intent.

3 MusicRecoIntent Dataset

Our corpus is based on *MusicRecoNER* (Epure and Hennequin, 2023), which contains English-language music recommendation requests collected from Reddit[2]. These requests, wrote by users for other users, have not been corrected or standardized, which explains their variable quality. They are mostly open-ended with their length varying according to the level of detail and complexity. For the purpose of the present study, only one third of the dataset was retained, namely 2,291 user queries.

Manual Annotation. The annotation task aimed to label all musical descriptors in each query according to whether the user wanted them, wanted something similar, or wanted to avoid them. Two annotators were instructed to annotate descriptors in the categories introduced in Section 1, together with their corresponding preference-bearing intent.

[2] https://www.reddit.com/

Validation. We measured inter-annotator agreement separately for descriptor extraction and preference-bearing roles. Cohen's Kappa is unsuitable for descriptor identification, which is multi-label and span-based. Thus, agreement was computed at the span level as a percentage. This metric naturally accounts for multiple descriptors per query and variations in annotated elements. For the preference-bearing intents, we relied on Cohen's Kappa. Results are reported in Table 1.

	Aggr. (%)	Pref. intent (Kappa)	# Descriptors in Common
Decade	86.7	0.889	92
Genre	78.2	0.634	716
Instrument	66.1	1.000	253
LC	66.8	1.000	160
Mood	69.4	0.898	410
NE	85.4	0.752	1677
Country	81.7	1.000	53
Global	77.1	0.927	3361

Table 1: Inter-annotators Agreements per Category

Overall, the agreement on descriptor extraction is substantial, with a global rate of 0.771. Agreement varies across categories, ranging from 0.661 for *Instrument* to 0.867 for *Decade*. The main disagreements are often due to typographical variations (e.g., *oppressive / oprressive*), spontaneous normalization or correction (e.g., *1980s / 80s*), segmentation differences (e.g., *frank oceans blonde / frank ocean, blonde*), or context-driven additions or omissions (e.g., *klassische / klassische musik*). The annotators noted the absence of a category dedicated to more structural musical information, such as rhythm or song composition.

The preference-bearing intent agreement was computed on descriptors extracted by both annotators, and is generally high: 0.927, indicating a strong agreement. Category-level Kappa scores range from 0.634 for *Genre* to 1 for *Instrument*, *Listening Context* (LC), and *Country*. Most descriptors were marked as desired (positive affinity), while named entities were frequently marked as referential, rather than being explicit targets. Negative preferences are quite rare across all categories.

Final Dataset Overview. To create the final dataset, the annotators reviewed all points of disagreement together and reached a consensus on each annotation. The final corpus contains a total of 3,935 annotations. Table 2 provides a detailed overview of descriptors per category, indicating whether they were annotated with a positive (+), negative (-), or referential (~) preference-bearing role. On average, each query contains between 1 and 3 elements per category.

	#Descr.	+	~	-	Coverage
Decade	94	84	9	1	3.03
Genre	792	749	25	18	3.37
Instru.	352	337	1	14	1.81
LC	217	215	2	0	1.12
Mood	551	539	0	12	2.13
NE	1870	253	1613	4	1.10
Country	59	57	2	0	1.59
Total	3935	2234	1652	49	1.51

Table 2: Descriptive Statistics of the Corpus by Category

4 Automatic Annotation with LLMs

To extract descriptors and preference-bearing intents, we rely on Ollama[3], an open-source framework that facilitates the use of a wide range of LLMs. We experiment with a diverse set of models and sizes: Gemma 3 (4B, 12B, 27B) (Gemma Team et al., 2025), LLaMA 3 (1.8B) (Grattafiori et al., 2024), Mistral (7B) (Jiang et al., 2023), and Qwen 3 (8B, 32B) (Yang et al., 2025). We design a single prompt that instructs the model to identify music entities, as well as the other descriptor types, using a broad definition to avoid imposing too many restrictions. The full prompt is provided in Appendix B.1. To guide the model and enhance consistency and accuracy, the task was illustrated with six concrete examples. A second prompt was employed to determine the preference-bearing intent of each descriptor. This subsequent prompt is provided in Appendix B.2.

Table 3 summarizes the number of descriptors extracted by the tested models. Models generally produced more descriptors than those in the manually annotated dataset. To ensure reliability and limit hallucinations, only descriptors similar enough to the original text were retained. The preference-bearing intent was predicted afterward for the descriptors generated by Gemma3:27b, the model with the best performance on the initial task.

Evaluation Metrics. The most common evaluation metrics used in the extraction of musical entities are precision, recall, and F-score. We extend these metrics to better analyze the types of errors our system may produce, following the methodology proposed by Batista (2018). For each prediction, we classify it as correct, missing, spurious, incorrect, or partial (overlaps partially with the expected entity). Then, results are evaluated for exact match only (a prediction is considered correct if its segmentation is exact) and partial match (A prediction is considered correct if its segmentation is at least partially accurate, i.e. if it has at least one word in common with the expected prediction). It should be noted that these scores depend both on annotation quality and on model performance.

LLM	Exact	Partial	# Descr.
Gemma3:4b	0.66	0.74	4405
Gemma3:12b	0.68	0.75	5010
Gemma3:27b	**0.69**	**0.76**	4860
LLaMA3.1:8b	0.60	0.68	4535
Mistral:7b	0.56	0.64	4813
Qwen3:8b	**0.69**	**0.76**	4628
Qwen3:32b	0.67	0.73	5034

Table 3: Overall F1-scores (Exact and Partial) and Number of Descriptors Extracted by Different LLMs.

Overall Results. Table 3 shows the performance of various LLMs. Gemma (12b, 27b) and Qwen 8b achieve the strongest overall performance, with the highest exact and partial F1-scores. In contrast, Llama 8b and Mistral 7b perform less consistently across descriptor categories. Overall, larger or more recent architectures tend to generalize better at this extraction task. Based on this, we selected Gemma 27B (which was faster than Qwen3-8B) for a detailed analysis and further experiments.

Descriptor categories	**Exact Match**			**Partial Match**		
	P	R	F1	P	R	F1
NE	0.86	0.82	0.84	0.93	0.88	0.90
Genre	0.83	0.80	0.82	0.92	0.88	0.90
Mood	0.85	0.78	0.82	0.93	0.85	0.89
LC	0.47	0.42	0.45	0.74	0.65	0.69
Instrument	0.69	0.62	0.66	0.85	0.76	0.80
Country	0.96	0.88	0.92	0.98	0.90	0.94
Decade	0.87	0.83	0.85	0.93	0.89	0.91
Overall	0.62	0.77	0.69	0.69	0.85	0.76

Table 4: Precision, Recall and F1-scores with Gemma3:27B

Results on Descriptors Extraction. Tables 4 shows that NEs are well recognized, with an Exact Match F1-score of 0.84. The model accurately identifies explicit and well-defined descriptors, such as Country, which achieves a F1-score above 0.90. Decade, Genre, and Mood are also well extracted. The comparison between Exact Match and Partial Match illustrates the effect of segmentation tolerance, boosting F1-score from 0.69 to 0.76. This indicates that the model often identifies descriptors correctly, but token boundaries are imperfect.

The relatively lower scores for Listening Context (LC) and Instrument are partly due to limitations

[3] https://ollama.com/

in the manual annotation process. LC often involves detailed descriptors, leading to annotation variability (e.g., "*I want music reco, in the summer my dad and me are doing a trip to norway*" - *[summer, norway]* vs. *['dad', 'summer', 'trip to norway']*), while Instrument performance is limited by segmentation issues (e.g., *electric guitar* vs. *guitar*). These factors explain the gap between Exact and Partial Match scores, highlighting the difficulty of achieving consistent span annotations, whether human- or LLM-generated.

Results on Preference-Bearing Intent Prediction. For this evaluation, we consider only the descriptors annotated both manually and by Gemma3:27B (3,030 descriptors). Among these descriptors, 89% of LLM predictions matched the ground-truth. Table 5 summarizes the results. Overall, the confusion matrix shows that the model performs well, though it tends to overpredict positive preferences at the expense of referential cases—231 cases are incorrectly predicted as positive. Although negative preferences are accurately extracted, their low frequency limits the robustness of the conclusions.

True \ Pred	+	–	~	#Descr.
+	0.94	0.01	0.05	1627
–	0.00	0.90	0.10	31
~	0.17	0.01	0.82	1372

Table 5: Normalized Confusion Matrix (per True Class) for Preference-Bearing Intent Prediction with Gemma 3:27B.

Qualitative analysis. When comparing manual annotations with Gemma-3 predictions (cf. Table 6), several recurring sources of disagreement emerge. In many cases, both the human annotator and the model identify the same underlying information but disagree on segmentation—some merge multiple tokens into a single descriptor, others split them into separate units. Boundary disagreements also arise from differences in expected granularity: predicted spans may be shorter or longer than the ground-truth, often reflecting annotation guideline ambiguities rather than genuine model errors.

Additional discrepancies involve truncated or slightly altered descriptors, particularly for named entities. These cases generally reflect minor lexical drifts rather than semantic misunderstanding. The model also tends to over-annotate highly generic musical terms (e.g., *music, song, album*), which, while domain-relevant, do not serve as meaningful descriptors. Omissions are also common, especially for song titles and artist names, whose surface forms often resemble ordinary text, making them difficult to distinguish from non-descriptive content. The scores in Table 4 illustrate the impact of these discrepancies. Exact match penalizes any deviation—including segmentation differences—whereas partial match tolerates boundary variations. Segmentation disagreements are pronounced for Listening Context and Instrument descriptors, partly due to guideline limitations that did not anticipate certain edge cases, explaining some MusicRecoIntent boundary inconsistencies.

For the second annotation task—determining the preference-bearing role of a descriptor as positive, negative, or referential—additional sources of disagreement arise. The most frequent occur when similarity requests are interpreted as positive preferences, particularly when phrasing includes "*like*" questioning the prompt design. Interestingly, nearly 80% of these cases involve named entities. Less common but more challenging are sentences with strong negation (e.g., *hate, not, unless*) while expressing an overall positive intent; in these cases, half of the errors concern musical genres. Examples include: "*I hate most rap but want to get more into the genre*" or "*Don't listen to a lot of EDM but I really like Porter Robinson's Shelter.*"

Finally, a subset of disagreements stems from genuine ambiguity, where multiple interpretations are possible. For example, "*more songs like 1000 Rounds by Pouya and Ghostemane*" may refer either to the song alone or to the song and its artists, and both readings are plausible.

5 Conclusion

By introducing *MusicRecoIntent*, a corpus annotated with musical descriptors and preference-bearing roles, this work provides a benchmark for fine-grained modeling of user intent in music-related queries. Our results show that LLMs reliably capture explicit, well-defined descriptors—such as named entities, country, genre, and mood, and predict preference-bearing roles with high accuracy for positive and negative cases, while referential roles are more challenging and often overpredicted. Beyond model performance, our analysis reveals shared challenges for both annotators and LLMs, particularly around boundary decisions, granularity, and semantic ambiguity, suggesting that future improvements will require clearer annotation guidelines and prompts.

References

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. 2023. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*.

David S. Batista. 2018. Named-entity evaluation metrics based on entity-level.

Mathieu Delcluze, Antoine Khoury, Clémence Vast, Valerio Arnaudo, Léa Briand, Walid Bendada, and Thomas Bouabça. 2025. Text2playlist: Generating personalized playlists from text on deezer. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part V*, page 164–170, Berlin, Heidelberg. Springer-Verlag.

SeungHeon Doh, Minz Won, Keunwoo Choi, and Juhan Nam. 2023. Toward universal text-to-music retrieval. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Elena Epure and Romain Hennequin. 2023. A human subject study of named entity recognition in conversational music recommendation queries. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1281–1296, Dubrovnik, Croatia. Association for Computational Linguistics.

Elena V. Epure and Romain Hennequin. 2022. Probing pre-trained auto-regressive language models for named entity typing and recognition. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1408–1417, Marseille, France. European Language Resources Association.

. Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Raghav Gupta, Renat Aksitov, Samrat Phatale, Simral Chaudhary, Harrison Lee, and Abhinav Rastogi. 2023. Conversational recommendation as retrieval: A simple, strong baseline. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pages 155–160, Toronto, Canada. Association for Computational Linguistics.

Simon Hachmeier and Robert Jäschke. 2024. Information extraction of music entities in conversational music queries. In *Proceedings of the 3rd Workshop on NLP for Music and Audio (NLP4MusA)*.

Simon Hachmeier and Robert Jäschke. 2025. A benchmark and robustness study of in-context-learning with large language models in music entity detection. In *Proceedings of the 31th International Conference on Computational Linguistics*.

Christine Hosey, Lara Vujović, Brian St. Thomas, Jean Garcia-Gathright, and Jennifer Thom. 2019. Just give me what i want: How people use and evaluate music search. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA. Association for Computing Machinery.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jinsung Kim, Seonmin Koo, and Heui-Seok Lim. 2025. Semantic inversion, identical replies: Revisiting negation blindness in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21445–21482.

Alessandro B. Melchiorre, Elena V. Epure, Shahed Masoudian, Gustavo Escobedo, Anna Hausberger, Manuel Moussallam, and Markus Schedl. 2025. Just ask for music (jam): Multimodal and personalized natural language music recommendation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys '25, page 615–620. ACM.

Enrico Palumbo, Marcus Isaksson, Alexandre Tamborrino, Maria Movin, Catalin Dincu, Ali Vardasbi, Lev Nikeshkin, Oksana Gorobets, Anders Nyman, Poppy Newdick, Hugues Bouchard, Paul Bennett, Mounia Lalmas, Dani Doro, Christine Doig Cardet, and Ziad Sultan. 2025. You say search, i say recs: A scalable agentic approach to query understanding and exploratory search at spotify. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys '25, page 1117–1121, New York, NY, USA. Association for Computing Machinery.

Lorenzo Porcaro and Horacio Saggion. 2019. Recognizing musical entities in user-generated content. *Computación y Sistemas*, 23(3):1079–1088.

Abhinaba Roy, Renhang Liu, Tongyu Lu, and Dorien Herremans. 2025. Jamendomaxcaps: A large scale music-caption dataset with imputed metadata. *arXiv preprint arXiv:2502.07461*.

Rebecca Salganik, Teng Tu, Fei-Yueh Chen, Xiaohao Liu, Kaifeng Lu, Ethan Luvisia, Zhiyao Duan, Guillaume Salha-Galvan, Anson Kahng, Yunshan Ma, and Jian Kang. 2025a. Musicsem: A dataset of music descriptions on reddit capturing musical semantics. In *Late-Breaking Demo, 26th Conference of the International Society for Music Information Retrieval (ISMIR 2025)*.

Rebecca Salganik, Teng Tu, Fei-Yueh Chen, Xiaohao Liu, Kaifeng Lu, Ethan Luvisia, Zhiyao Duan, Guillaume Salha-Galvan, Anson Kahng, Yunshan Ma, and Jian Kang. 2025b. Musicsem: A semantically rich language-audio dataset of organic musical discourse. In *AI for Music Workshop, 39th Conference on Neural Information Processing Systems (NeurIPS 2025)*.

Bruno Sguerra, Marion Baranes, Romain Hennequin, and Manuel Moussallam. 2022. Navigational, informational or punk-rock? an exploration of search intent in the musical domain. In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '22, page 202–211, New York, NY, USA. Association for Computing Machinery.

Tereza Vrabcová, Marek Kadlčík, Petr Sojka, Michal Štefánik, and Michal Spiegel. 2025. Negation: A pink elephant in the large language models' room? *Preprint*, arXiv:2503.22395.

Benno Weck, Holger Kirchhoff, Peter Grosche, and Xavier Serra. 2024. *WikiMuTe: A Web-Sourced Dataset of Semantic Descriptions for Music Audio*, page 42–56. Springer Nature Switzerland.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

A Disagreements Examples in the Dataset

Table 6 shows examples of disagreements observed between manual annotations and Gemma3:27b annotations.

Query	Manual annotations	Gemma3:27b annotations	Disagreements Type
songs similar to japanese ceremonial tea	['japanese ceremonial tea']	['japanese', 'ceremonial', 'tea']	Segmentation
more genre busting like wu tang vs beatles	['wu tang vs beatles']	['wu tang', 'beatles']	Segmentation
songs like kendrick lamars maad city	['kendrick lamars', 'maad city']	['kendrick lamars maad city']	Segmentation
nostalgic indie pop alt rock songs	['nostalgic', 'indie', 'pop', 'alt rock']	['nostalgic', 'indie pop', 'alt rock']	Segmentation
looking for a wedding first dance song	['wedding first dance song']	['wedding', 'dance']	Segmentation
music similar to these songs do you feel it chaos chaos	['do you feel it', 'chaos chaos']	['chaos chaos']	Omission
looking for lovechild of metallica	['lovechild', 'metallica']	['metallica']	Omission
songs based off of eminems phenomenal	['eminems', 'phenomenal']	['eminem', 'phenomenal']	Truncation
similar to otherside by the red hot chili peppers	['the red hot chili peppers', 'otherside']	['red hot chili peppers', otherside']	Truncation
music genre for grandiose stylistic trumpet	['grandiose stylistic trumpet']	['music', 'grandiose', 'stylistic', 'trumpet']	Over-detection + Segmentation
a sad song	['sad']	['sad', 'song']	Over-detection

Table 6: Examples of disagreements between manual annotations and Gemma3:27b annotations

B Prompts

B.1 Prompt for Musical Descriptors Extraction

> You're an assistant specialized in music. Your aim is to detect and extract musical descriptors mentioned in sentences. We want to extract all types of musical descriptors cited in the given text: artist names (or group/band names) and all musical work of art as song title or album title, musical genre, decade, location, mood, listening context, instruments, etc. Output must be a list that can contains all the descriptors found. Here some examples of the output expected:
> - eg 1 : text : "I love rock and roll" - output: ['rock and roll'],
> - eg 2 : text : "I love sad spanish love songs " - output: ['sad', 'spanish', 'love'].
> - eg 3 : text : "je veux du gros rap des années 80 pour faire la fête" - output: ['rap', 'années 80', 'fête'],
> - eg 4 : text : "I want some french songs like j'irai ou tu iras - Céline dion and JJ Goldman" - output: [french, "Céline dion", "JJ Goldman", "j'irai ou tu iras"].
> - eg 5 : text : "Pop music to cook and sing" - output: ['pop', 'cook', 'sing].
> - eg 6 : text : "I love Whenever, Wherever - Shakira " - output: ["Shakira", "Whenever, Wherever"].
>
> Do not explain what you are doing, do not add any information that is not in the text to process and do not modify or correct the extracted text. Write only the output in one line. If you don't find descriptors, write [].</Task>
> Please extract the musical descriptors cited in the following text : "sentence"

B.2 Prompt for Preference-Bearing Intent Extraction

You're an assistant specialized in natural langage processing. You will receive: a user query, and a list of descriptors that appear in the query. For each descriptor, determine the user's intention toward it using the following labels:
"'+' : the user is explicitly looking for this descriptor.
"'~' : the user is looking for something similar or related, but not necessarily exactly this descriptor.
"'-' : the user wants to exclude this descriptor (indicated by negations such as no, not, without, avoid, exclude, etc.).
Output format is strict: Return only a Python list of tuples of the form: [(descriptor, intention), ...]. No explanations, no extra text.
Rules:
a) Assign '+' if the descriptor is explicitly mentioned in the query as something the user wants, without being negated or rejected.
b) Assign '-' if the query explicitly negates, excludes, rejects (e.g., no X, not X, without X, exclude X, avoid X, etc.).
c) Assign '~' if the descriptor is mentioned in a way that suggests the user is looking for something related, similar, or loosely connected, but not exactly that descriptor.
Here some examples of the output expected:
- eg 1: User query: 'dark 90s music' ; Descriptors: '['dark', '90s']' -> Expected output: [('dark', '+'), ('90s', '+')]
- eg 2: User query: 'music like Abba but more rock' ; Descriptors: '['Abba', 'rock']' -> Expected output: [('Abba', '~'), ('rock', '+')]
- eg 3: User query: 'rock music without guitar' ; Descriptors: '['rock', 'guitar']' -> Expected output: [('rock', '+'), ('guitar', '-')]
- eg 4: User query: 'Celine dion song without Goldman' ; Descriptors: '['Celine dion', 'Goldman']' -> Expected output: [('Celine dion', '+'), ('Goldman', '-')]
- eg 5: User query: 'Calm rock song similar to the beatles' ; Descriptors: '['Calm ', 'rock', the beatles]' -> Expected output: [('Calm', '+'), ('rock', '+'), ('the beatles', '~')]
Now process the following instance: User query: ¨sentence;̈ Descriptors: "desc"